Dr. Bright

CANADA

DEPARTMENT OF MINES AND TECHNICAL SURVEYS

MINES BRANCH

OTTAWA

MINERAL SCIENCES DIVISION

INTERNAL REPORT MS-65-29

THE USE OF A PDP-5 COMPUTER ON MÖSSBAUER EXPERIMENTS AT DIGITAL EQUIPMENT OF CANADA LIMITED, CARLETON PLACE, ONTARIO

by

R.H. Goodman

Physics and Radiotracer Subdivision

February 15, 1965

Mineral Sciences Division Internal Report MS-65-29

THE USE OF A PDP-5 COMPUTER ON MÖSSBAUER EXPERIMENTS AT DIGITAL EQUIPMENT OF CANADA LIMITED, CARLETON PLACE, ONTARIO

by

R.H. Goodman*

- - - -

In a previous report (MS-64-93), the author discussed a visit to Digital Equipment of Canada Limited (D.E.C.) and described a small general-purpose computer, the PDP-5. After some consideration of its possibilities, it was decided to evaluate this machine for the Mössbauer experiments currently in progress at the Mines Branch. A complete set of apparatus for Mössbauer experiments was transported to D.E.C. and an experiment was performed. This report includes a description of the computer, the input equipment, the program and the results.

---

* Scientific Officer, Physics and Radiotracer Subdivision, Mineral Sciences Division, Mines Branch, Department of Mines and Technical Surveys, Ottawa, Canada.

THE PDP-5 COMPUTER

The D.E.C. PDP-5 is a small general-purpose computer with a storage capacity of 4096 (2^{12}) 12-binary bit words. This storage is divided into 32, 128 word pages, of which the current page and page 0 are directly addressable. In such a binary computer, it is easier to use an octal notation rather than a decimal notation. Each page therefore contains $(0\text{-}177)_8$ locations and there are $(0\text{-}7777)_8$ registers available for a total of $(0\text{-}37)_8$ pages. Some of these registers can be used for special purposes as can be seen from the following table:

Address	Purpose
0_8	Program counter
1_8	PI storage
2_8	First instruction after interrupt
10_8-17_8	Auto index register.

Also, there is a loading routine which is generally stored in locations 7700 to 7777. The rest of the machine is free for programming and storage.

Another feature of this computer is the availability of direct access to the accumulator through the program-interrupt facility (P.I.). This allows direct electrical signals from experimental apparatus to be put into the memory of the computer, to be processed and to be typed out.

The P.I. is turned ON by the statement ION and OFF by IOF. When the P.I. is ON, a signal on pin 26 of a 50-pin connector (J101 on the

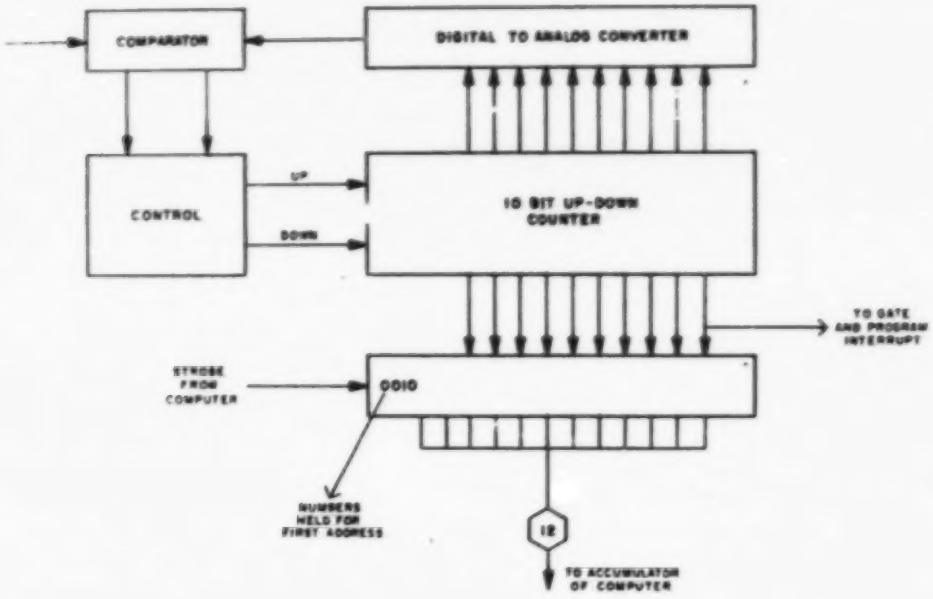
computer) causes the computer to stop the program in progress and start in storage location 2. The contents of the program counter are stored in location 1 so that at the end of the interrupt, the program will start where it was interrupted. Since many external devices could generate at the P.I. signal, location 2 generally contains a routine to search for the correct device. If there are many devices, location 2 will contain a jump command so as not to use the auto index registers for simple programming. The P.I. will cause the computer to generate pulses which are used to read or strobe external counters and transfer their contents to the accumulator. This P.I. facility takes some time; each instruction requires 12 microseconds to perform, and hence the total time may be 150 or more microseconds per program interrupt event. A summary of interconnections and programming instructions is included in Appendix 1.

THE INPUT EQUIPMENT

To obtain a Mössbauer spectrum, the velocity of the source versus the counting rate must be plotted. The velocity is measured using an LVsyn velocity transducer, the output of which must be converted to a digital number before it is fed into the computer. This was accomplished with a continuous ADC; the general scheme of this conversion process is shown in Figure 1.

FIGURE 1

Block Diagram of Continuous(Analog to Digital Converter)

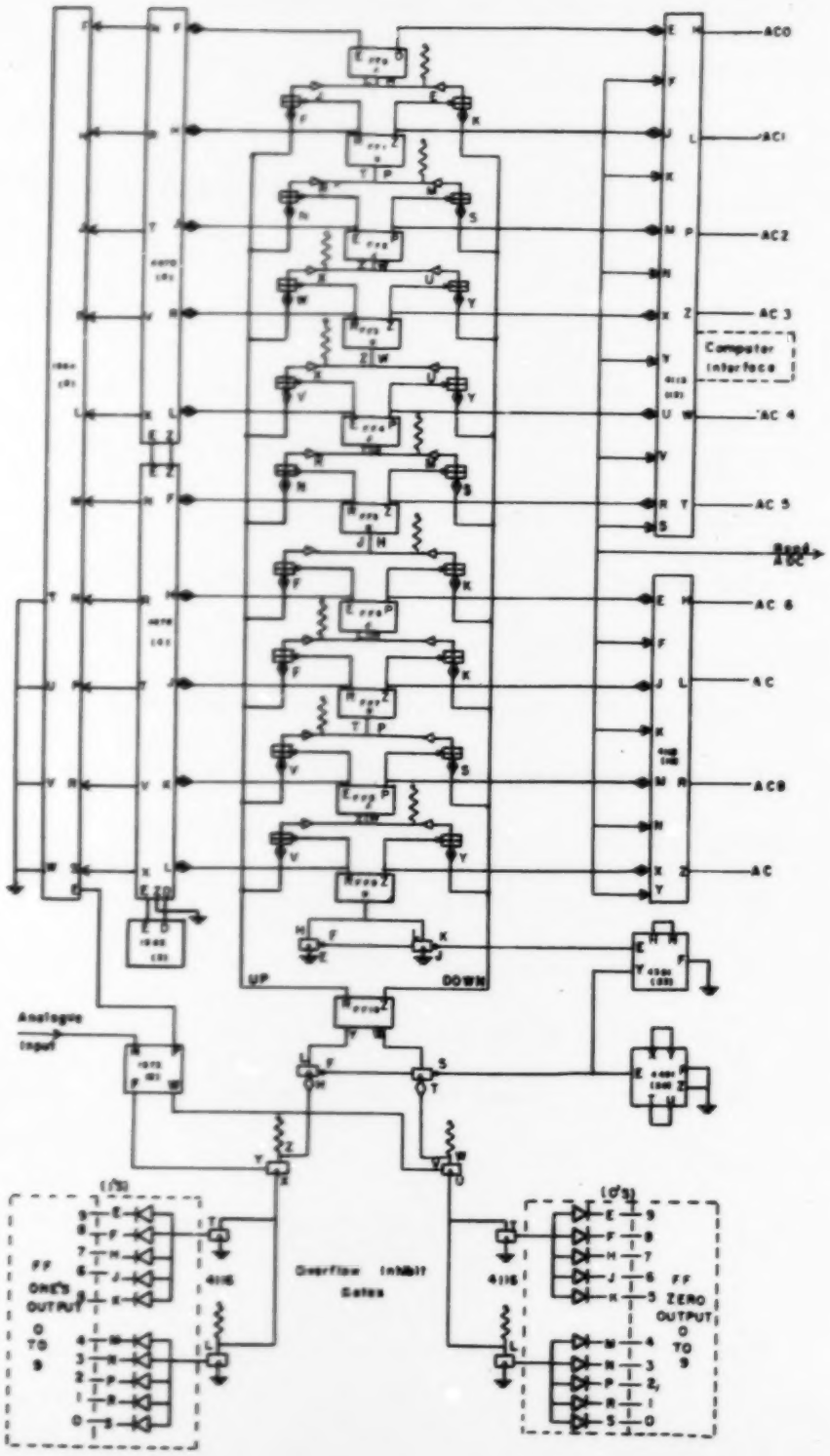


The operation of such an ADC is straightforward. A comparator produces a 1 or 0 level depending on the difference between the output of the digital-to-analog converter and the analog input signal. If the input is greater than the digital-to-analog signal, the control circuit causes the counter to count up until the comparator inputs are of equal value. The first stage of the counter will oscillate back and forth searching for the digital number equivalent to the input signal. The number on the counter thus will be proportional to the velocity of the source at any given time. A 10-binary bit device of which only the eight most significant bits were read was used to increase differential linearity. The least significant bit (LSB) of these eight is used to trip the program interrupt of the computer.

The complete circuit for the ADC is shown in Figure 2. The gates are used to read the output of the converter when a signal (READ ADC 6702) is generated by the computer.

The counts for the velocity interval are recorded on another 12-bit register and are read by the command READ COUNTER 6704. The counter is cleared by another pulse (6712) and the counts are again accumulated for a new velocity interval. The program sequence for this is shown in Appendix 2. The main program is concerned with the live time display of data on the oscilloscope and the accumulation of data from the experiment. The display procedure is interrupted on the requirement to accept data and started again after the data accepting program has been executed. There is a complication in that the total capacity of a register is 12 binary bits, that is 4096, and thus double precision or two registers must be used to store a significant amount of data. This is accomplished by the use of a link. The link is a 13th bit on the accumulator, and is used to indicate overflow of the accumulator. The total program is also shown in Appendix 2. By a suitable combination of hardware (modules) and software (programs), a system was built to collect data and display it. A typical module is shown in Figure 3. These modules may be regarded as true black boxes and can be used as building blocks for more advanced systems. They operate on logic levels of 0 and -3 volts and generally use negative input pulses.

FIGURE 2


AC0
AC1
AC2
AC 3
Computer Interface
AC 4
AC 5
Read AC
AC 6
AC
AC8
AC
UP
DOWN
Analogue Input
Overflow Inhibit Gates
FF ONE'S OUTPUT 0 TO 9
FF ZERO OUTPUT 0 TO 9
4116

EXPERIMENTAL ARRANGEMENT

A block diagram of the electronics is shown in Figure 4. The source is detected by a proportional counter as shown in Figure 5. The feedback control consisted of a number of operational amplifiers arranged in a manner similar to that described by Cohen et al (1). The output of the LVsyn was amplified and fed into the continuous ADC. The output detector was analyzed by a pulse-height analyser. Events with an energy corresponding to 14 keV were transferred to the 12-bit register for reading into the computer. The complete system is shown in Figure 6.

FIGURE 3

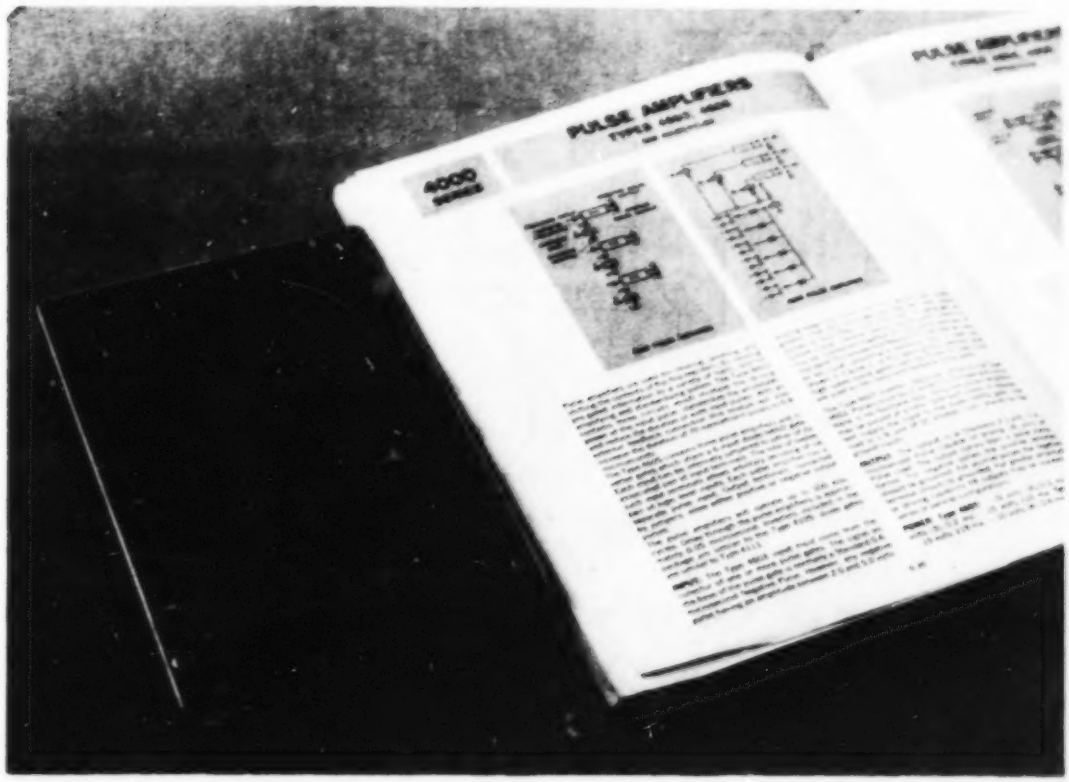


FIGURE 4

Block Diagram of Electronics

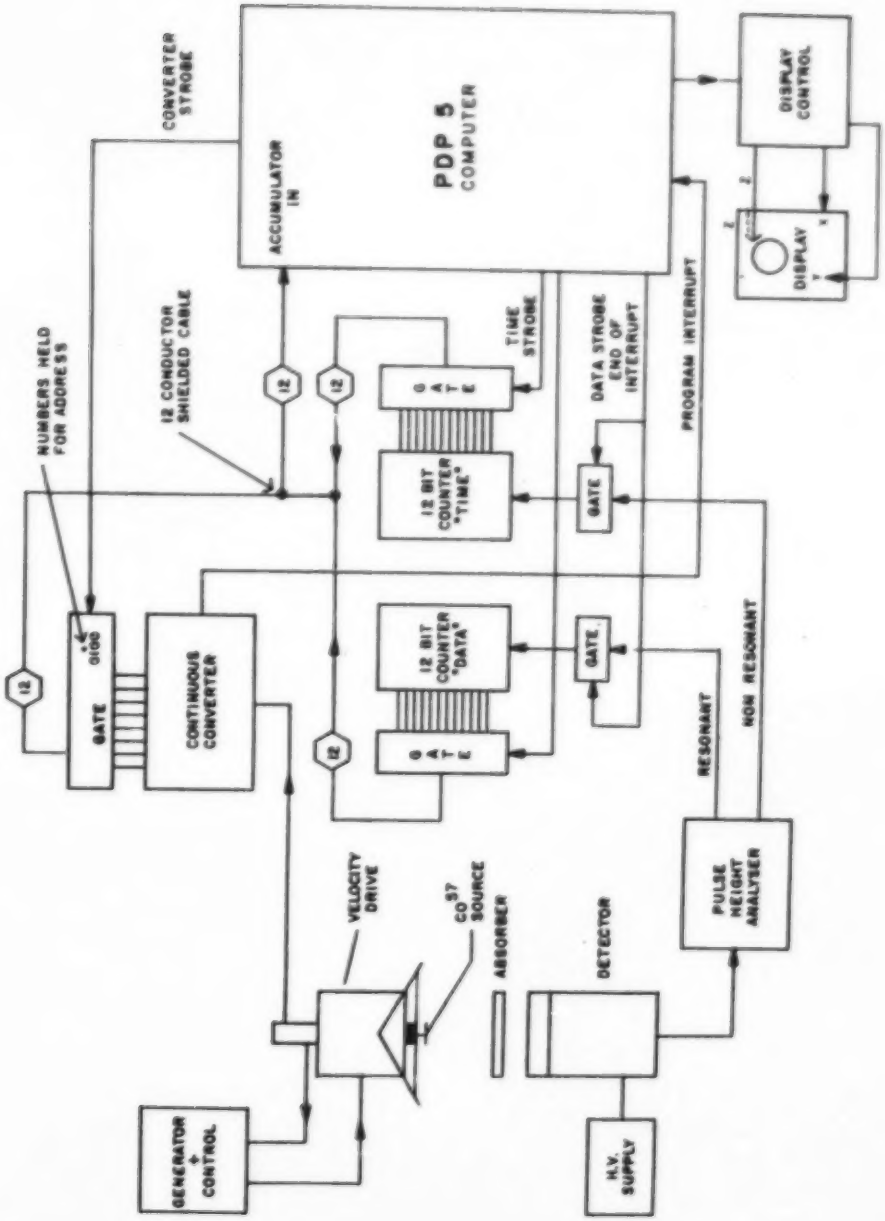


FIGURE 5

Photograph of Apparatus

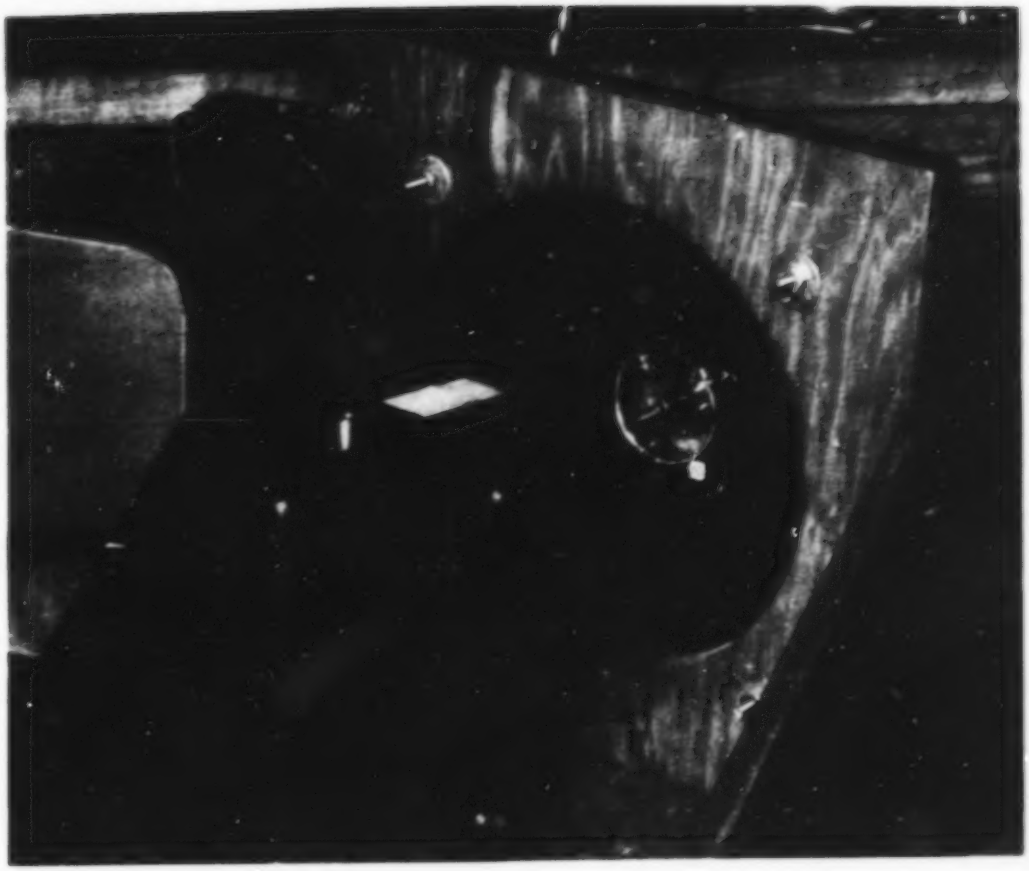
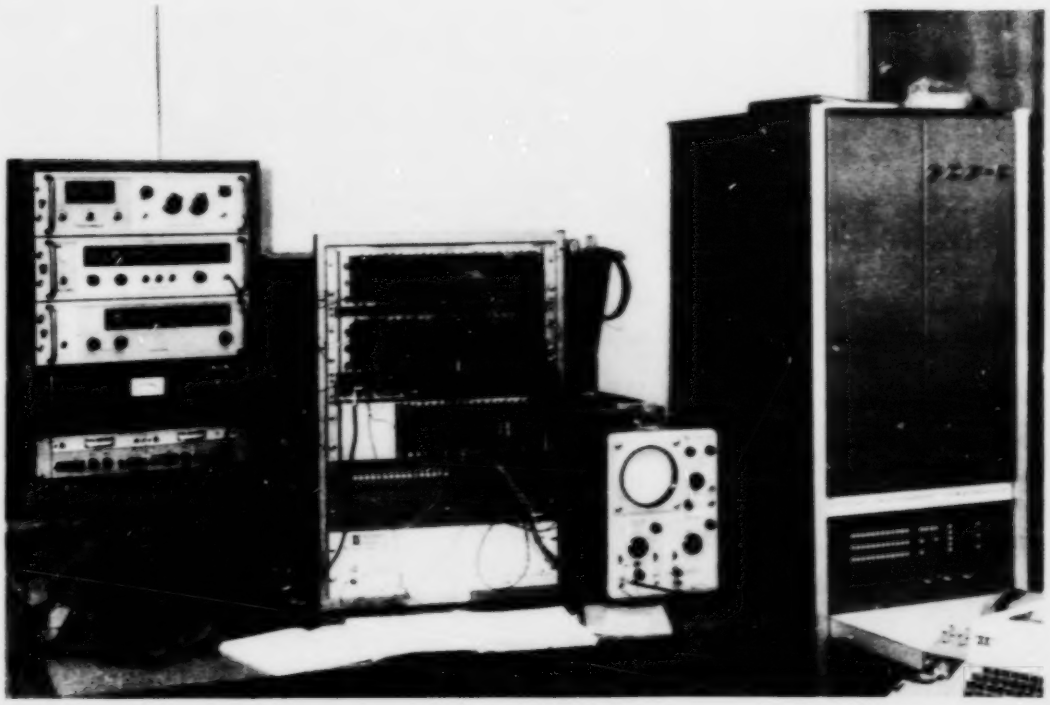


FIGURE 6

Experimental Procedure



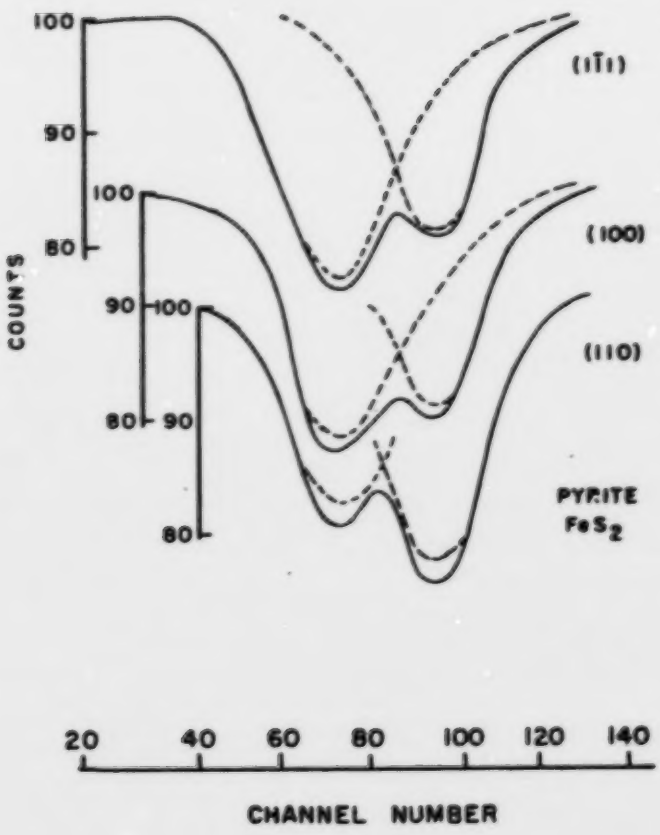
The system was calibrated for velocity using the well known iron spectrum (2). Approximately 3×10^{-5} counts per channel were accumulated over a period of some 14 hours. This spectrum was found to agree with that of other workers.

Samples of FeS_2 (pyrite) were prepared from naturally occurring single crystals. Thin slices cut in the 100, 110 and 111 directions were mounted on glass slides and used as absorbers. The Goldanskii Effect (3) was observed and can be clearly seen from the data as shown in Figure 7.

FIGURE 7

Mössbauer Spectrum of Pyrite

Further experiments will be required to interpret this data.



ACKNOWLEDGEMENTS

The work, including the building of the ADC, was completed in about two weeks. This would not have been possible without the willing co-operation of many people at Digital Equipment of Canada Limited. Messrs. D.J. Doyle and J. Muzeneek designed the ADC and were always ready to discuss problems which were encountered. Mr. J. Richardson wrote the program and assisted in setting up the experiment. The help and encouragement of Mr. William MacGregor is gratefully acknowledged. Every phase of the project involved him in some way and all problems were tackled with vigour and enthusiasm.

REFERENCES

1. R.L. Cohen, P.G. McMullen and G.K. Wertheim, Rev.Sci.Inst. 34 671-673 (1963).
2. R.S. Preston, S.S. Hanna and J. Heberle, Phys.Rev. 128 2207 (1962).
3. V.I. Goldanskii, E.F. Makarov and V.V. Khrapor, Phys.Lit. 3 344 (1963).

RHG:ST

APPENDIX 1

INSTRUCTIONS

MEMORY REFERENCE INSTRUCTIONS

Mnemonic Symbol	Operation Code	Time (μsec)	Operation
and Y	0	18	Logical AND. The AND operation is performed between the C(Y) and the C(AC). $C(Y)_i \wedge C(AC)_i \Rightarrow C(AC)_i$.
tad Y	1	18	Twos complement add. The C(Y) are added to the C(AC) in twos complement arithmetic. $C(Y) + C(AC) \Rightarrow C(AC)$.
isz Y	2	18	Index and skip if zero. The C(Y) are incremented by one in twos complement arithmetic. If the resultant C(Y) = 0, the next instruction is skipped. $C(Y) + 1 \Rightarrow C(Y)$. If result = 0, $C(PC) + 1 \Rightarrow C(PC)$.
dca Y	3	18	Deposit and clear AC. The C(AC) are deposited in core memory location Y and the AC is cleared. $C(AC) \Rightarrow C(Y)$, then $0 \Rightarrow C(AC)$.
jms Y	4	24	Jump to subroutine. The C(PC) are deposited in core memory location Y. The next instruction is taken from location Y + 1. $C(PC) + 1 \Rightarrow C(Y)$ $Y + 1 \Rightarrow C(PC)$
jmp Y	5	12	Jump to Y. The C(PC) are set to address Y. The next instruction is taken from core memory location Y. $Y \Rightarrow C(PC)$.

BASIC IOT MICROINSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation
		PROGRAM INTERRUPT
ion	6001	Turn interrupt on
iof	6002	Turn interrupt off
		ANALOG-TO-DIGITAL CONVERTER
adc	6004	Convert analog to digital

rar	7010	2	Rotate the C(AC) and the C(L) right one place. $C(AC)_i \Rightarrow C(AC)_{i-1}$ $C(AC)_{11} \Rightarrow C(L)$ $C(L) \Rightarrow C(AC)_0$
rtr	7012	2, 3	Rotate two right.
cml	7020	2	Complement L. $C(\overline{L}) \Rightarrow C(L)$
cma	7040	2	Complement AC. $C(\overline{AC}) \Rightarrow C(AC)$
cll	7100	1	Clear L. $0 \Rightarrow C(L)$
cla	7200	1	Clear AC. $0 \Rightarrow C(AC)$

GROUP 2 OPERATE MICROINSTRUCTIONS

Mnemonic Symbol	Octal Code	Event Time	Operation
hlt	7402	3	Halt. Stops the program.
osr	7404	3	OR with Switch Register C(SR) V C(AC) $\Rightarrow$ (CAC)
skp	7410	1	Skip, unconditional. C(PC) + 1 $\Rightarrow$ C(PC)
snl	7420	1	Skip on non-zero L. If C(L) = 1, then C(PC) + 1 $\Rightarrow$ C(PC)
szl	7430	1	Skip on zero L. If C(L) = 0, then C(PC) + 1 $\Rightarrow$ C(PC)
sza	7440	1	Skip on zero AC. If C(AC) = 0, then C(PC) + 1 $\Rightarrow$ C(PC)
sna	7450	1	Skip on non-zero AC. If C(AC) $\neq$ 0, then C(PC) +1 $\Rightarrow$ C(PC)
sma	7500	1	Skip on minus AC. If $C(AC)_0 = 1$, then C(PC) + 1 $\Rightarrow$ C(PC)
spa	7510	1	Skip on positive AC. If $C(AC)_0 = 0$, then C(PC) + 1 $\Rightarrow$ C(PC)
cla	7600	2	Clear AC $0 \Rightarrow C(AC)$

GROUP 1 OPERATE MICROINSTRUCTIONS

Mnemonic Symbol	Octal Code	Event Time	Operation
nop	7000	—	No operation. Causes a 12 μsec program delay.
iac	7001	3	Index AC. C(AC) +1 $\Rightarrow$ C(AC)
ral	7004	2	Rotate the C(AC) and the C(L) left one place. $C(AC)_i \Rightarrow C(AC)_{i-1}$ $C(L) \Rightarrow C(AC)_{11}$ $C(AC)_0 \Rightarrow C(L)$
rtl	7006	2, 3	Rotate two left.

HIGH SPEED PERFORATED TAPE READER

rsf	6011	Skip if Photoreader flag = 1
rrb	6012	Read the contents of the photoreader buffer into $C(AC)_{28-35}$ and clear the Photoreader flag
rfc	6014	Clear Photoreader flag and buffer, fetch one character from tape and load it into the photoreader buffer, and set the Photoreader flag when done.

HIGH SPEED PERFORATED TAPE PUNCH

psf	6021	Skip if High Speed Punch flag = 1
pcf	6022	Clear the Punch flag and buffer
ppc	6024	Load the punch buffer from $C(AC)_{28-35}$ and punch the character (this instruction does not clear the High Speed Punch flag or buffer).
pls	6026	Clear the Punch flag and buffer, load the punch buffer from $C(AC)_{28-35}$, punch the character, and set the Punch flag when done.

TELETYPE KEYBOARD/READER

ksf	6031	Skip if Keyboard flag = 1
kcc	6032	Clear AC and Keyboard flag.
krs	6034	Read the contents of the keyboard buffer into $C(AC)_{28-35}$ (does not clear AC or flag.)
krb	6036	Clear AC, read keyboard buffer into AC, clear Keyboard flag.

TELETYPE TELEPRINTER/PUNCH

tsf	6041	Skip if Teleprinter flag = 1
tcf	6042	Clear Teleprinter flag
tls	6046	Load the LUO from the $C(AC)_{28-35}$, clear Teleprinter flag, and print and/or punch the character.

OSCILLOSCOPE DISPLAY AND PRECISION CRT DISPLAY

dcx	6051	Clear X buffer
dxl	6053	Clear and load the X buffer $C(AC)_{2-17} \Rightarrow C(YB)$.
dcy	6061	Clear Y buffer
dyl	6063	Clear and load the Y buffer $C(AC)_{2-17} \Rightarrow C(YB)$.
dix	6054	Intensify the point defined by C(XB) and C(YB)
diy	6064	Intensify the point defined by C(XB) and C(YB)
dxs	6057	Executes the combined functions of dxl followed by dix
dys	6067	Executes the combined functions of dyl followed by diy.
dsf	6071	Skip if Display flag = 1
dcf	6072	Clear Display flag
dlb	6074	Load brightness register. $C(AC)_{9-17} \Rightarrow C(BR)$

INTERFACE CONNECTIONS

Connection points for interface signals at the I/O connectors 1J01 and 1J02 are listed in the following table.

PDP-5 I/O CONNECTIONS

Signal	Symbol	Terminal	Signal	Symbol	Terminal
AC $\frac{1}{0}$ Out		1	AC $\frac{1}{4}$ In		17
AC $\frac{1}{1}$ Out		2	AC $\frac{1}{5}$ In		18
AC $\frac{1}{2}$ Out		3	AC $\frac{1}{6}$ In		19
AC $\frac{1}{3}$ Out		4	AC $\frac{1}{7}$ In		20
AC $\frac{1}{4}$ Out		5	AC $\frac{1}{8}$ In		21
AC $\frac{1}{5}$ Out		6	AC $\frac{1}{9}$ In		22
AC $\frac{1}{6}$ Out		7	AC $\frac{1}{10}$ In		23
AC $\frac{1}{7}$ Out		8	AC $\frac{1}{11}$ In		24
AC $\frac{1}{8}$ Out		9	IOS		25
AC $\frac{1}{9}$ Out		10	PI		26
AC $\frac{1}{10}$ Out		11	MB $\frac{0}{3}$		27
AC $\frac{1}{11}$ Out		12	MB $\frac{1}{3}$		28
AC $\frac{1}{0}$ In		13	MB $\frac{0}{4}$		29
AC $\frac{1}{1}$ In		14	MB $\frac{1}{4}$		30
AC $\frac{1}{2}$ In		15	MB $\frac{0}{5}$		31
AC $\frac{1}{3}$ In		16	MB $\frac{1}{5}$		32
MB $\frac{0}{6}$		33	IOP 4*		43
MB $\frac{1}{6}$		34	IOP 4		44
MB $\frac{0}{7}$		35	1 MC clock		45
MB $\frac{1}{7}$		36	I/O Hlt		46
MB $\frac{0}{8}$		37	AC Clear		47
MB $\frac{1}{8}$		38	Restart		48
IOP 1*		39	Power Clear		49
IOP 1		40	Ground		50
IOP 2*		41			
IOP 2		42			

*Ground side of pulse transformer secondary winding.

CODES

TELETYPE CODE

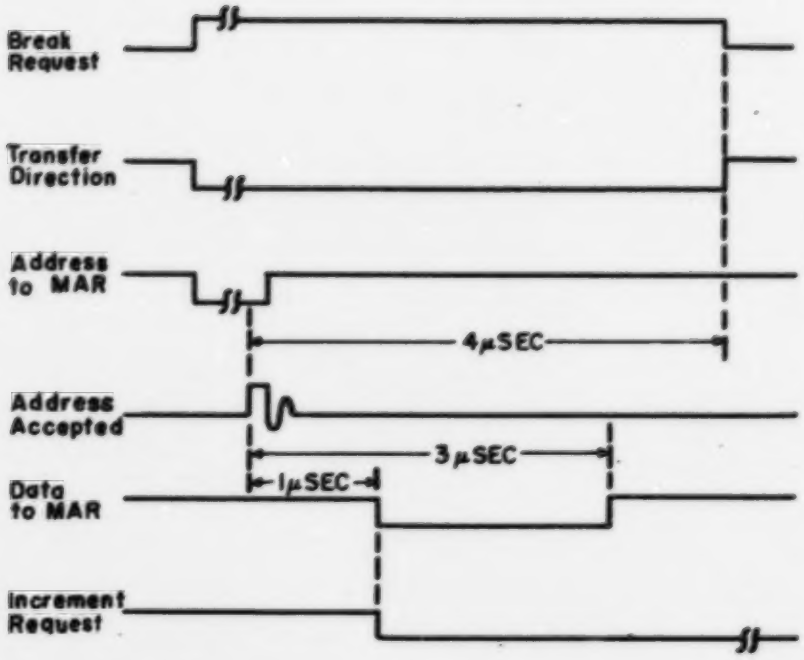
Character	8-Bit Code (in octal)	6-Bit Trimmed Code (in octal)	Character	8-Bit Code (in octal)	6-Bit Trimmed Code (in octal)
A	301	01	!	241	41
B	302	02	"	242	42
C	303	03	#	243	43
D	304	04	$	244	44
E	305	05	%	245	45
F	306	06	&	246	46
G	307	07	'	247	47
H	310	10	(	250	50
I	311	11	)	251	51
J	312	12	*	252	52
K	313	13	+	253	53
L	314	14	,	254	54
M	315	15	-	255	55
N	316	16	.	256	56
O	317	17	/	257	57
P	320	20	:	272	72
Q	321	21	;	273	73
R	322	22	<	274	74
S	323	23	=	275	75
T	324	24	>	276	76
U	325	25	?	277	77
V	326	26	@	300	00
W	327	27	[	333	33
X	330	30	\	334	34
Y	331	31	]	335	35
Z	332	32	↑	336	36
0	260	60	←	337	37
1	261	61	EOT	204	—
2	262	62	W RU	205	—
3	263	63	RU	206	—
4	264	64	BELL	207	—
5	265	65	Line Feed	212	—
6	266	66	Return	215	—
7	267	67	Space	240	40
8	270	70	ACK	374	—
9	271	71	ALT MODE	375	—
			Rub Out	377	—

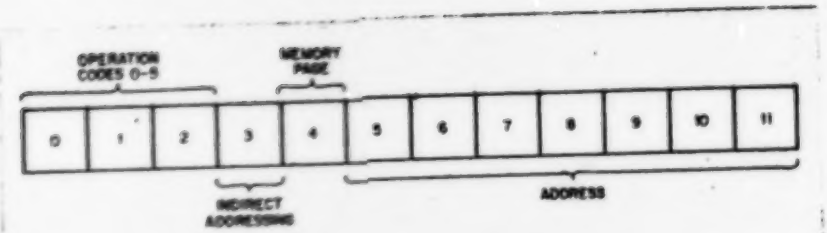
Connection points for data break signals at connector 1J03 are presented in the following table.

PDP-5 DATA BREAK CONNECTIONS

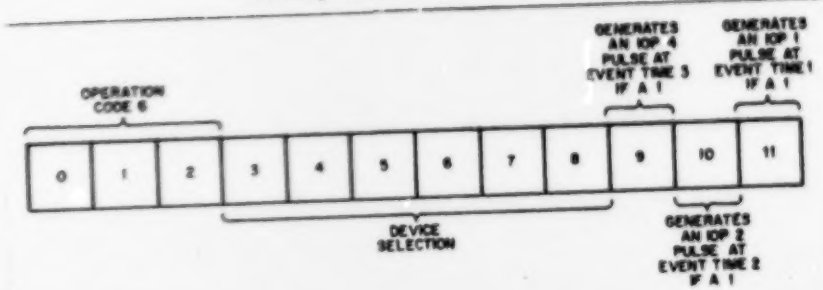
Signal	Symbol	Terminal	Signal	Symbol	Terminal
MB $\frac{1}{0}$ Out	◇	1	MB $\frac{1}{11}$ In	◇	24
MB $\frac{1}{1}$ Out	◇	2	Data Addr. Bit 0	◇	26
MB $\frac{1}{2}$ Out	◇	3	Data Addr. Bit 1	◇	27
MB $\frac{1}{3}$ Out	◇	4	Data Addr. Bit 2	◇	28
MB $\frac{1}{4}$ Out	◇	5	Data Addr. Bit 3	◇	29
MB $\frac{1}{5}$ Out	◇	6	Data Addr. Bit 4	◇	30
MB $\frac{1}{6}$ Out	◇	7	Data Addr. Bit 5	◇	31
MB $\frac{1}{7}$ Out	◇	8	Data Addr. Bit 6	◇	32
MB $\frac{1}{8}$ Out	◇	9	Data Addr. Bit 7	◇	33
MB $\frac{1}{9}$ Out	◇	10	Data Addr. Bit 8	◇	34
MB $\frac{1}{10}$ Out	◇	11	Data Addr. Bit 9	◇	35
MB $\frac{1}{11}$ Out	◇	12	Data Addr. Bit 10	◇	36
MB $\frac{1}{0}$ In	◇	13	Data Addr. Bit 11	◇	37
MB $\frac{1}{1}$ In	◇	14	Break' State	◆	41
MB $\frac{1}{2}$ In	◇	15	Run' State	◆	42
MB $\frac{1}{3}$ In	◇	16	Break Request	◇	43
MB $\frac{1}{4}$ In	◇	17	Transfer Direction (Into PDP-5)	◆	44
MB $\frac{1}{5}$ In	◇	18	Increment Request	◆	45
MB $\frac{1}{6}$ In	◇	19	SP 0	▶	46
MB $\frac{1}{7}$ In	◇	20	Power Clear	▶	47
MB $\frac{1}{8}$ In	◇	21	Data = > MB	▷	48
MB $\frac{1}{9}$ In	◇	22	Address Accepted	▷	49
MB $\frac{1}{10}$ In	◇	23	Ground	⏚	50

Signal for Data Break Cycle

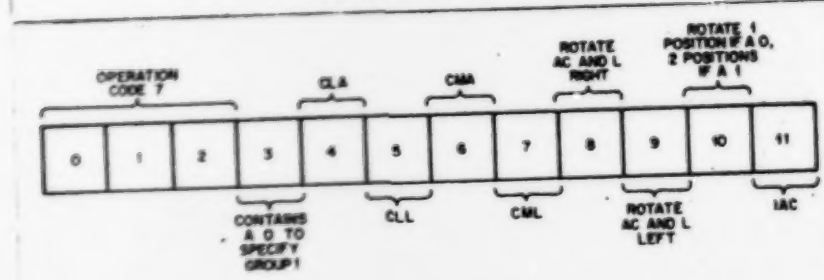




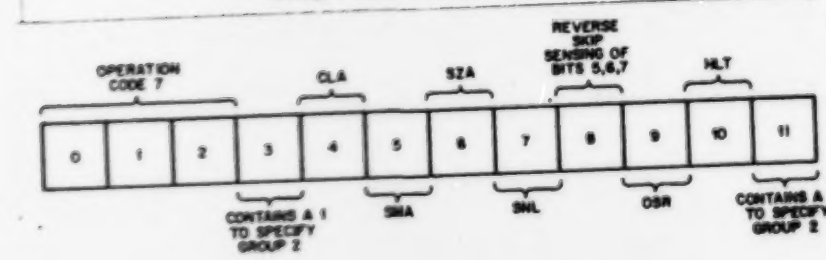
Memory Reference Instruction Bit Assignments



IOT Instruction Bit Assignments



Group 1 Operate Instruction Bit Assignments



Group 2 Operate Instruction Bit Assignments

APPENDIX 2

This appendix gives the program used for Mössbauer experiments in connection with the D.E.C. PDP-5 computer. The starting address is 200 for clearing all locations; 213 for accumulating over present data; 214 for display without accumulation. The switch register is set to the low order address for display and type-out. For both display and type-out, 256 channels are used. Type-out starts at 400, and normalization at 5000. Storage locations are 1000-1377 for low order data, 1400-1777 for high order data; 2000-2377 and 2400-2777 for the clock, and 3000-3777 for the ratio of these two. The following is the program.

```
*6000
DDVD,  0
       DCA DIV A
       TAD SET
       DCA L2
       TAD I DIV A
       DCA HORD
       ISZ DIV A
       TAD I DIV A
       DCA LORD
       ISZ DIV A
       TAD I DIV A
       DCA QUOH
       ISZ DIV A
       TAD I DIV A
       DCA QUOL
       TAD HORD
       SMA
       JMP GO
       ISZ L2
       CMA CLL
       DCA HORD
       TAD QUOL
       CI A
       DCA QUOL
       TAD QUOH
```

```
        CMA
        SNL
        JMP DARN
        I AC CLL
        DCA QUOH
        SNL
        JMP DARN+1
        TAD LORD
        CMA I AC CLL
        DCA LORD
        SZL
        ISZ HORD
GO,     CLL CLA
        TAD I DDVD
        DCA DIV A
        TAD I DIV A
        DCA AC 1 H
        ISZ DIV A
        TAD I DIV A
        DCA AC 1 L
        TAD AC 1 H
        SMA CLA
        JMP XX 10
        ISZ L2
        NOP
        JMP XX 11
XX 10,  TAD AC 1 L
        CMA I AC
        DCA AC 1 L
        TAD AC 1 H
        CMA
        SZL
        I AC
        DCA AC 1 H
XX 11,  TAD MIF
        DCA DV1
        CLL
        JMP DV2
DV3,    TAD LORD
        RAL
        DCA LORD
        TAD HORD
        RAL
        DCA HORD
        TAD LORD
        TAD AC 1 L
```

```
        DCA SAV
        RAL
        TAD HORD
        TAD AC 1 H
        SZL
        JMP OK
        CL A
DV2,    TAD QUOL
        RAL
        DCA QUOL
        TAD QUOH
        RAL
        DCA QUOH
        ISZ DV1
        JMP DV3
        TAD QUOL
        ISZ L2
        JMP XX 8
        CMA I AC
        DCA QUOL
        TAD QUOH
        CMA
        SZL
        I AC
XX 7,   ISZ DDVD
        JMP I DDVD
XX 8,   DCA QUOL
        TAD QUOH
        JMP XX 7
OK,     DCA HORD
        TAD SAV
        DCA LORD
        JMP DV2
DARN,   DCA QUOH
        TAD LORD
        DMA
        DCA LORD
        JMP GO
SAV,    0
DIV A,  0
HORD,   0
LORD,   0
QUOH,   0
QUOL,   0
AC 1 H, 0
AC 1 L, 0
L2,     0
```

```
SET,  776
MIF,  705
DV 1,  0
```

Normalizing Routine

```
*5000
NORL, CLA CLL
      TAD REGS          span of data
      DCA PERT
      TAD 1K            first address of data
      DCA ALOD
AGNI, TAD ALOD
      TAD 4D
      DCA AHOD
      TAD ALOD
      TAD 1K
      DCA ALDV
      TAD ALDV
      TAD 4D
      DCA AHDV
      TAD I AHOD
      DCA 1 DIV         arranging divisor and dividend in proper
      TAD I ALOD                    order
      DCA 2DIV
      DCA 3DIV
      DCA 4DIV
      TAD I AHDV
      DCA 1VID
      TAD I ALDV
      DCA 2VID
      TAD ALDV
      TAD 1K
      DCA ALOQ
      TAD ALOQ          storage of answer
      TAD 4D
      DCA AHOQ
      TAD ALBI
      JMS I DDV I       jump to divide routine at 6000
      1 VID
      DCA I AHOQ
      TAD I QUOI
      DCA I ALOQ
      ISZ ALOD
      ISZ PERT
      JMP AGNI
      JMP I SHE         jump to start of program
```

```
1K,     1000
4D,     400
ALOD, 0
AHOD, 0
ALDV, 0
AHDV, 0

*5100
1DIV,  0
2DIV,  0
3DIV,  0
4DIV,  0
1VID,  0
2VID,  0
DDV I, DDVD
QUOI,  QUOL
SHE,   BEGN
REGS,  0-377
PERT,  0
ALOQ,  0
AHOQ,  0
ALBI,  5100
```

24-Bit Binary to Decimal Conversion and Type-Out

```
*600
DBND, 0
        CLA CLL
        TAD I DBND
        DCA GET
        TAD I GET
        SPA
        CML
        CLA
        TAD PLUS
        SZL
        TAD MNS
        JMS PRNT
        TAD I GET
        CLL
        SPA
        CMA CML
        DCA HIGH
        TAD LOOP
        DCA CNT
        TAD ADDR
        DCA Z 0010
        ISZ GET
```

```
        ISZ DBND
        TAD I GET
        SZL
        CMA CLL I AC
        SZL
        ISZ HIGH
        DCA LOW
ARND,   TAD I Z 0010
        DCA HSUB
        JMS DO
        ISZ CNT
        JMP ARND
        TAD PLUS
        JMS PRNT
        JMP I DBND
DO,     0
        TAD I Z 0010
        DCA I SUB
        CLL
        TAD I SUB
        TAD LOW
        DCA TEML
        RAL
        TAD HSUB
        TAD HIGH
        SPA
        JMP OUT
        ISZ BOX
        DCA HIGH
        TAD TEML
        DCA LOW
        JMP DO+3
OUT,    CLA
        TAD BOX
        JMS PRNT
        DCA BOX
        JMP I DO
PRNT,   0
        TAD TWO
        TLS
        TSF
        JMP .-1
        CLA
        JMP I PRNT
LOOP,   7771
CNT,    0
```

print out

```
HIGH,  0
LOW,   0
ADDR,  CON1-1
TWO,   0260
PLUS,  7760
MNS,   0015
HSUB,  0
LSUB,  0
BOX,   0
TEML,  0
GET,   0
CON 1, 7413
       6700
       7747
       4540
       7775
       4360
       7777
       6030
       7777
       7634
       7777
       7766
       7777
       7777
```

Output Control Routine

```
*400
OUPT,  CLA CLL
       JMS TCR
       LAS                 address of first type-out in switch register
       DCA ADR 1
       DCA CHAN
       TAD ADR 1
       DCA OCNT
       TAD OCNT
       TAD 4 C             end of type-out
       DCA HOCT
       TAD M 4HN
       DCA DONE
TOP 1, CLA CLL
       TAD M 5
       DCA DCNR
       TAD CHAN
       JMS I SBNI          channel print-out
```

```
        JMS TSP
        JMS TSP          spaces
        JMS TSP
TOP 2,  TAD I OCNT
        DCA HOLD+1
        TAD I HOCT
        DCA HOLD
        ISZ OCNT
        ISZ HOCT
        ISZ CHAN
        JMS I DBNI       number print-out
        HOLD
        ISZ DONE
        SKP
        JMP GOHM
        ISZ DCNR
        JMP .+3          number-per-line control
        JMS TCR
        JMP TOP 1
        JMS TSP
        JMP TOP 2
GOHM,   JMS TCR
        JMP I FINL       to program start
CHAN,   0
ADR 1,  1000
OCNT,   0
4 C,    400
HOCT,   0
M 4HN,  0-400
DONE,   0
M 5,    0-5
DCNR,   0
SBNI,   SBNC
HOLD,   0
HOLD + 1,  0
DBNI,   DBND
FINL,   BEGN
```

Type Carriage Return

```
TCR,    0
        CLA
        TAD TTAB + 2
        JMS OTY
        TAD TTAB + 3
        JMS OTY
        DCA TBC
        JMP I TCR
```

```
/TYPE A SPACE
TSP,  0
      CLA
      TAD TTAB + 4
      JMS OTY
      JMP I TSP
/TYPE A TABULATION
TYT,  0
      CLA
      TAD TBC
      TAD TTAB
      SMA
      JMP .-2
      DCA TBC
      JMS TSP
      JMP .-1
OTY,  0
      TLS
      TSF                    output
      JMP .-1
      CLA
      ISZ TBC
      JMP I OTY
      JMP I TYT
TEM,  0
TEMY, 0
TBC,  0
STBC, 0
FLAK, 0
CH 2, 0
TTAB, 7770
      77
      215
      212
      240
      17
      7766
      260
      7740
      100
      SPA
```

12-Bit Binary to Decimal Conversion and Type-Out

```
*260
SBNC, 0
      CLL
      SPA
      CML
      DCA VAL
      DCA OBX
      TAD CNTR
      DCA I CNT
      TAD ARDD
      DCA XYZ + 2
      TAD SULP
      SZL
      TAD SMN
      JMS I PNRT
      TAD VAL
      SPA
      CMA I AC
XYZ,  DCA VAL
      TAD VAL
      TAD CON
      SMA
      ISZ OBX
      SMA
      JMP XYZ
      CLA
      TAD OBX
      JMS I PNRT
      DCA OBX
      ISZ XYZ + 2
      ISZ I CNT
      JMP XYZ + 1
      JMP I SBNC
ARDD, TAD CON
OBX,  0
I CNT, 0
VAL,  0
CNTR, 7774
CON,  6030
      7634
      7766
      7777
```

```
PNRT, PRNT
SULP, 7760
SMN,  0015
```

Program Interrupt Routine

```
*1
        0000            storage of program counter
        DCA ACCM  ]
        RAL       ]     store previous program
        DCA LINK  ]
        6 701           skip on flag
        JMP DATA
        JMP DATA

*20

DATA,   6702            read ADC
        DCA TEM 1       store ADC
        6 704           read cnts
        DCA TEM 2       store cnts
        6 713           read clock, clear and start
        DCA TEM 3       store clock
        TAD TEM 1  ]    start address at 1000
        TAD 1 TH   ]
        DCA ADD 1       cnts address
        TAD ADD 1
        TAD 1 TH
        DCA ADD 2       clock address
        TAD I ADD 1     previous cnts
        TAD TEM 2       new cnts
        DCA I ADD 1     restore
        SZL             skip on zero link
        JMP DPCT        double precision cnts
1 DAT,  CLA CLL
        TAD I ADD 2     clock address
        TAD TEM 3
        DCA I ADD 2
        SZL
        JMP DPCL        double precision clock
        JMP EXIT
DPCT,   TAD ADD 1
        TAD 4 HUN
        DCA ADD 3       address of high order cnts
        ISZ I ADD 3
        JMP 1 DAT
DPCL,   TAD ADD 2
        TAD 4 HUN
```

```
        DCA ADD 4     address of high order clock
        ISZ I ADD 4
        JMP EXIT
EXIT,   KCC           ]
        TCF           ] clear flags
        6714
        TAD LINK      }
        RAR CLL       } restore program
        TAD ACCM      }
        ION           interrupt ON
        JMP I 1       back to program
ACCM,   0000          ]
LINK,   0000          |
ADD 1,  0000          |
ADD 2,  0000          |
ADD 3,  0             | storage
ADD 4,  0             |
TEM 1,  0             |
TEM 2,  0             |
TEM 3,  0             |
1 TH,   1000          |
4 HUN,  400           ]
```

Clearing and Display Routine

```
*200

BEGN,   HLT
CLER,   CLA CLL
        IOF
        TAD 1 THO     first address
        DCA TEMP
        TAD FULL
        DCA STEP      tally counter
        DCA I TEMP    clear
        ISZ TEMP
        ISZ STEP
        JMP CLER + 6
        ION           turn on interrupt
DSLY,   CLA CLL
        DCA AXIS
        LAS           switch register for first address of display
        DCA 1 ADD
        TAD 1 ADD
        DCA STOR
        TAD TALY      set width of span
        DCA SPAN
```

```
STRT, TAD AXIS
      DXL          X axis
      CLA CLL
      TAD I STOR
      NOP
      NOP
      NOP          Y axis
      DYS
      CLA CLL
      ISZ AXIS
      ISZ STOR
      ISZ SPAN
      JMP STRT
      JMP DSLY
1 THO, 1000
FULL, 0-3777
1 ADD, 1000
TALY, 0-377
TEMP, 0000          storage
STEP, 0000
STOR, 0000
SPAN, 0000
AXIS,  0
$
```

THE FOLLOWING ARE THE ADDRESSES USED IN THIS PROGRAM

Divider

```
DDVD  6000
DIVA  6156
SET   6166
L2    6165
HORD  6157
LORD  6160
QUOH  6161
QUOL  6162
GO    6045
DARN  6150
AC1H  6163
AC1L  6164
XX10  6063
XX11  6073
MIF   6167
DV 1  6170
DV 2  6116
DV 3  6077
SAV   6155
```

OK	6144
XX 8	6141
XX 7	6137
NORL	5000
REGS	5111
PERT	5112
1 K	5050
ALOD	5052
AGNI	5005
4 D	5051
AHOD	5053
ALDV	5054
AHDV	5055
1 DIV	5100
2 DIV	5101
3 DIV	5102
4 DIV	5103
1 VID	5104
2 VID	5105
ALOQ	5113
AHOQ	5114
ALBI	5115
DDDVI	5106
QUOI	5107
SHE	5110

24-Bit Output Routine

DBND	600
GET	716
PLUS	710
MNS	711
PRNT	673
HIGH	704
LOOP	702
CNT	703
ADDR	706
LOW	705
ARND	635
HSUB	712
DO	645
LSUB	713
TEML	715
OUT	666
BOX	714
TWO	707
CON 1	717

Output Control

OUPT	400
TCR	466
ADR 1	451
CHAN	450
OCNT	452
4 C	453
HOCT	454
M 4HN	455
DONE	456
TOP 1	414
M 5	457
DCNR	460
SBNI	461
TSP	476
TOP 2	424
HOLD	462
DBNI	464
GOHM	446
FINL	465
TTAB	532
OTY	514
TBC	526
TYT	503
TEM	524
TEMY	525
STBC	527
FLAK	530
CH 2	531

12-Bit Output

SBNC	260
VAL	323
OBX	321
CNTR	324
1 CNT	322
ARDD	320
XYZ	301
SULP	332
SMN	333
PNRT	331
CON	325

Program Interrupt

ACCM 72
LINK 73
DATA 20
TEM 1 100
TEM 2 101
TEM 3 102
1 TH 103
ADD 1 74
ADD 2 75
DPCT 50
1 DAT 41
DPCL 55
EXIT 62
4 HUN 104
ADD 3 76
ADD 4 77

Program Start

BEGN 200
CLER 201
1 THO 242
TEMP 246
FULL 243
STEP 247

Display

DSLY 214
AXIS 252
1 ADD 244
STOR 250
TALY 245
SPAN 251
STRT 224